

The 'Essence' of Agile Metrics



I've been struggling for quite some time figuring out the "essence of" agile metrics. What are good ones versus bad ones? Are there related sets that are relevant and how many are needed? And how much change do we need to make from our traditional metrics thinking towards agile metrics?

Another thought I've had is whether they should represent a trailing indicator or a leading indicator? Meaning should we measure the processes (at the beginning or leading edge) or should we more so focus on the results & learning (at the end or from the trailing edge)?

For example—

I think of measuring estimate accuracy (estimate + actual) as a leading indicator. Why is that since you measure actuals at the end of your work?

I'm thinking of it from the point of view of what you're trying to improve. In this case, your measuring estimates and actuals to improve your overall planning and estimation processes. These activities are typically front-loaded actions and are often static...meaning they are completed once at the beginning of the project.

The other challenge with this metric is taking corrective action for improvement. Effectively you can only effect a change with the next planning interval for a project. Depending on the length of that interval, improvements could be infrequent and delayed. Or depending on project success, they might not be useful at all.

So I've seen some agile teams that try to measure various things. Let's take a typical example—

How about if we measure user story volatility in the sprint? What we're trying to focus on is how many times do stories change (get added, become reframed or reduced in scope, or remove entirely) within any particular sprint. To what end?

Well, we'd like to improve our story grooming process and monitor how the team is delivering on their sprint planning commitments.

Ah, there's a key – sprint planning is sort of a leading indicator, in this case you're measuring the quality of story entry. What if we turn this into a trailing, results oriented, measurement? Let's measure the velocity of the team as they exit the sprint. We can measure velocity in stories produced and in story points delivered.

An even better trailing metric is to measure how the team delivered value towards their sprint goals. Meaning, did they deliver what they committed to? Not in tasks or in stories, but in content that exactly met the spirit of their customers needs and the level of their commitment.

You notice that I reframed the notion of a trailing metric to a results oriented metric. Another way of thinking about it is measuring at the “outbound side” of the teams work. That way, you're leaving the doing of the work up to the team and simply measuring results or outcomes.

An example of why this is better.

Let's go back to the example I gave above. What is wrong with measuring story volatility? On the surface it appears fine. We can see how well the Product Owner staged the sprint and also how well the team is interpreting stories for execution. For example, if they “signed up” for 10 stories, but only delivered 7 because of story (requirement) volatility, then they have a lot of improvement to do...right? Well, that could be the case.

But it could also be the case that the team trimmed those stories because they were superfluous to the customers' needs. Meaning they were essentially waste. Or another positive interpretation of it could be the team became very creative and figured out a way to meet the spirit of the sprint goal with fewer stories. In fact, they may have even trimmed out some previous code in order to simplify the application—so the overall related LOC count might be quite low too.

Can you see the point I'm making? I find that leading indicator metrics can sometimes be more prone to metrics dysfunction than trailing, results oriented metrics.

In his book, *Measuring and Managing Performance in Organizations*, Robert Austin coined and explored the term metrics dysfunction. In simple terms, it's a metric whose collection influences the very behavior being measured—leading to metrics dysfunction. One of the classic examples of it is measuring software testers by the number of bugs they find. In this case, testers would report more trivial, non-germane bugs simply to hit the metric...and in the meantime they'd miss truly important bugs. Clearly not what the measurer intended!

In our example the leading metrics might influence the team towards not changing stories that need to be changed—simply to hit the metric of stories being consistent with the views from planning. I can see nothing more destructive to the agile mindset of the team than having metrics that drive negative behavior within the team—simply to meet or game the metrics.

So, what are some unhealthy leading metrics to avoid within agile teams?

- Planning quality – measuring how well you planned. An important part of this is change control, i.e. measuring how little or much change is ongoing.
- Requirement quality – measuring whether each requirements meets some sort of baseline definition or completeness. Or that each has a specific number of acceptance tests.
- Estimation quality – I’ve sort of beaten this down in the article—effectively anything that tries to measure estimation variance with actual effort.
- Arbitrary results – Counting LOC produced, or Bugs found, or Requirements written—virtually anything that is materially produced by the team where you negate the quality of the result and the application of common sense in that not all results need the same level of attention and thinking.

Conversely, what are some healthier trailing metrics to concentrate on within agile teams?

- Team agitation levels – capturing multi-tasking events, # of committed hours per sprint and variance from expected levels, recruiting support for team members
- Team velocity levels – trending over time, improvement trending (implying improved team dynamics), paying attention when team composition changes
- Impediment handling – #'s per team, avg. time to resolve, # that impacted the Sprint Goals
- Retrospective actions – is the team improving themselves based on retrospective results, how many improvements per sprint, average time to resolve
- Escapes per sprint – bugs found post-sprint, adherence levels to Done-ness Criteria

- Sprint Success / Failure – from the perspective of the Sprint Goal. Not so much focused at a Story or Task completeness level, but at an overall work delivered level

One of the things you'll notice with trailing indicator metrics is that there is an inherent real-time nature to them. We want to be sampling them (observing, discussing, driving action) each and every day during each iteration or sprint.

It's this level of team awareness of their performance (metrics) and active engagement towards continuous improvement that is a key for **functional metrics** in an agile context.